AMENDMENTS TO THE SPECIFICATION

In the Specification:

Please make the following amendments to the indicated paragraphs in the specification:

Page 9, last paragraph:

Referring now to Fig. 2, example framework objects 200 are illustrated in accordance with an aspect of the present invention. As noted above, one or more objects 200 can be supplied by a framework component that provide an interface between highly-threaded systems and lower-threaded modules. For example, at ~~240~~ 210 a Driver Frameworks Request Object can be generated or presented by the framework component. This object represents a request as managed by the driver frameworks, and various packages. A request may be passed to a driver by a package, and the driver can operate on the request by a set of exported methods. Driver Frameworks packages generally track a request, even if it's currently being operated on, or is "owned" by the driver due to an outstanding I/O that has been passed to the driver but not yet completed. A request object can be represented by a handle DFXREQUEST, for example.

Page 13, first paragraph:

Depending on the device driver's configuration, and the driver frameworks objects, event callbacks may be at DISPATCH or PASSIVE level. In order to properly support events that may occur at DISPATCH or PASSIVE level, presentation locks can be implemented with spinlocks and/or FAST_MUTEX's. Which lock is acquired can depend on the event's context, and the drivers configured threading model. In cases in which multiple objects may exist, such as for Device and IoQueue, API's for the presentation locks 320 generally act on the object instance, not the whole family of objects. So holding the presentation lock on one IoQueue does not prevent other IoQueue[[']]s from invoking their callback handlers. Depending on the driver's design and configuration, this may not be desirable. In this case one of the hierarchical locking models may be chosen which are described in more detail below.